Haaga-Helia
ammattikorkeakoulu Oy

# OpenLDAP keskitetty käyttöönotto SuSE Linux Enterprise -ympäristössä

Joel Lappalainen

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma
2016

Haaga-Helia
ammattikorkeakoulu Oy

# Tiivistelmä

**Tekijä(t)**
Joel Lappalainen

**Koulutusohjelma**
Tietojenkäsittelyn koulutusohjelma

**Opinnäytetyön otsikko**
OpenLDAP keskitetty käyttöönotto SuSE Linux Entreprise –ympäristössä

**Sivu- ja liitesivumäärä**
27 + 12

**Opinnäytetyön otsikko englanniksi**
Centralized deployment of LDAP in SuSE Linux enterprise environment

Ohjelmistojen keskitetty käyttöönotto ja hallinta ovat yleisiä toimintatapoja keskisuurten yritysten keskuudessa. Se helpoittaa useiden työpisteiden samanaikaista hallintaa, ja uusien työpisteiden käyttöönottoa. Projekti keskittyy keskitetyn käyttöjähallinnan OpenLDAP keskitettyyn käyttöönottoon uusilla työpisteillä.

Projekti kuvaa kuinka luodaan, testataan ja käytetään keskitetyn ohjelmistohallinnan ohjelmistoa Puppet, ja sen manifestien toimintaa SuSE Linux Enterprise ympäristössä. Projekti kattaa myös Puppet ohjelmiston käyttöönoton työpisteellä ja palvelimella.

SuSE Linux Enterprise 12 käyttöjärjestelmää käytettiin projektissa työpisteillä ja palvelinkoneella käyttöjärjestelmänä.

Projektissa toteutettiin toimiva keskitetyn käyttöönoton testiympäristö keskitetyn käyttäjähallinnan OpenLDAP varten SuSE Linux työpisteille.

Kaikki projektissa käytetyt konfiguroinnit ovat työn liitteenä, ja niitä saa vapaasti käyttää kuka tahansa.

Jatkotutkimusta varten, olisi hyödyllistä olla yksinkertainen dokumentaatio puppet työpisteen ohjelmiston keskitetystä levityksestä.

**Asiasanat**
Linux, SuSE, Puppet, LDAP, SSSD, PAM

Haaga-Helia
University of Applied Sciences

# Abstract

**Author(s)**
Joel Lappalainen

**Degree programme**
Bachelor of Business Administration in IT

| **Report/thesis title** | **Number of pages and appendix pages** |
| --- | --- |
| Centralized deployment of OpenLDAP in SuSE Linux enterprise environment | **27 + 12** |

Centralizing software deployment and management are common practices within medium sized companies. It eases management of multiple workstations, and deployment of new workstations. This project focuses on deployment of a centralized user-control OpenLDAP on new workstations.

This project describes and explains how to create, test and use centralized software management tool Puppet manifests in SuSE Linux Enterprise environment, as well as covers deployment of Puppet itself on the workstations.

The idea is represented through a fictional use-case, where an 11-step manual deployment process is being automatized to a single parameter.

SuSE Linux Enterprise 12 was used in this project to represent technology for server and client operating systems.

The result of this thesis was a successfully working test environment for centralized deployment of OpenLDAP centralized user-control on SuSE Linux workstations.

All of the necessary configuration files used in this project are attached to the thesis report, and are free to use by anyone.

As to further research objectives, a basic tutorial would be beneficial for the centralized deployment of the Puppet client software on workstations.

**Keywords**
Linux, SuSE, Puppet, LDAP, SSSD, PAM

# Sisällysluettelo


1 Johdanto ..... 1
2 Puppet ..... 2
2.1 Toiminta ..... 2
2.2 Manifestit, luokat ja resurssit ..... 4
2.3 Tietoturva ..... 4
3 OpenLDAP ..... 5
3.1 Toiminta ..... 5
3.2 Tietoturva ..... 6
4 SSSD – System Security Services Daemon ..... 7
5 Keskitetyn ohjelmistohallinnan toteutusosuus ..... 8
5.1 OpenLDAP ..... 8
5.1.1 Käyttötapaukset ..... 8
5.1.2 Selvitysosuus – palvelin ..... 8
5.1.3 Selvitysosuus - Päätekone ..... 9
5.2 Keskitetty ohjelmistonhallinta puppet ..... 17
5.2.1 Puppet-master asennus ja testaus SLES 12 palvelimella ..... 18
5.2.2 Puppet asennus ja testaus SLED 12 päätekoneella ..... 20
5.2.3 Arkkitehtuuri ..... 23
5.2.4 Testaus ja virheiden selvittelyt ..... 25
6 Pohdinta ..... 27
Lähteet ..... 28
Liitteet ..... 30
Liite 1. /etc/puppet/manifests/site.pp ..... 30
Liite 2. /etc/puppet/modules/sled12ldap/manifests/init.pp ..... 30
Liite 3. /etc/puppet/modules/sled12ldap/manifests/ldap_packages.pp ..... 30
Liite 4. /etc/puppet/modules/sled12ldap/manifests/ldap_configuration.pp ..... 31
Liite 5. /etc/puppet/modules/sled12ldap/manifests/ldap_services.pp ..... 33
Liite 6. /etc/puppet/modules/sled12ldap/files/nsswitch.conf ..... 33
Liite 8. /etc/puppet/modules/sled12ldap/files/pam/common-password ..... 36
Liite 9. /etc/puppet/modules/sled12ldap/files/pam/common-auth ..... 36
Liite 10. /etc/puppet/modules/sled12ldap/files/pam/common-account ..... 37
Liite 11. /etc/puppet/modules/sled12ldap/files/sssd/sssd.conf ..... 37
Liite 12. /etc/puppet/modules/sled12ldap/files/sudoers ..... 38

# 1 Johdanto

Järjestelmäkokonaisuuksien laajentuessa ympäristöjen ylläpidosta tulee haastavampaa, koska jokaiselle päätekoneelte tulisi konfiguroida organisaatiokohtaiset omat asetukset ja ylläpitää niitä, ettei tule käyttäjäpäällekkäysyyksiä tai käyttäjätunnuksia jäisi tekemättä. Toisekseen isommat muutokset esimerkiksi tietoturvapolitiikaan, voivat vaativat toimenpiteitä päätekoneilta ohjelmistojen puolesta. Nämä ovat haastavia ja aikaa vieviä toteuttaa manuaalisesti varsinkin suuremmissa organisaatioissa.

Tähän ratkaisuna on keskitetty ohjelmistonhallinta, jossa tieto vaadituista ohjelmista ja niiden konfiguroinneista pidetään palvelimella ja sieltä heijastetaan tai haetaan työpisteille suoraan ilman sen suurempia päätekonekohtaisia toimenpiteitä. Tämä tietenkin edellyttää hyvää suunnittelua ja toteutusta keskitetyltä hallinnalta, niin palvelimen kuin päätekoneiden kannalta.

Tämän projektin tarkoituksena on avata lukijalle keskitetyn käyttöönoton prosesseja, hyödyntäen fiktiivisen yrityksen valmiina olevaa toimintatapaa uusien koneiden käyttöönotossa. Projektissa toteutetaan LDAP autentikoinnin käyttöönotto päätekoneilla keskitetyn ohjelmistonhallinnan avulla.

Keskitetyn ohjelmiston- ja käyttäjänhallinnan palvelimena käytetään SuSE Linux Server 12 SP1 versiota, ja päätekoneena SuSE Linux Desktop 12 SP1 versiota.

Keskitettyyn ohjelmistonhallintaan käytetään Puppet 3.74 versiota, joka on viimeisin vakaa versio SuSE Linux Enterprise 12. Huomioitavaa on, että tämä ohjelma ei ole virallisesti SuSE Linux Enterprisen tukema, vaan kehitysrepositorioista haettu versio. Projektissa myös seurataan ja raportoidaan mahdollisista huomioiduista ongelmista tai virheistä.

Keskitettyyn käyttäjänhallintaan käytetään OpenLDAP 2.4.41 versiota, joka on SuSE Linux Server 12 SP1 asennusrepositoreissa vakiona mukana.

Projekti ei tule keskittymään LDAP palvelimen konfigurointiin, vaan LDAP autentikoinnin käyttöönottoon keskitetysti, jo olemassa olevalla keskitetyn käyttäjähallinnan järjestelmällä.

Työ koostuu kolmesta eri osuudesta: Teoriaosuudesta, käytännön toteutuksesta ja lopputuloksen pohdinnasta.

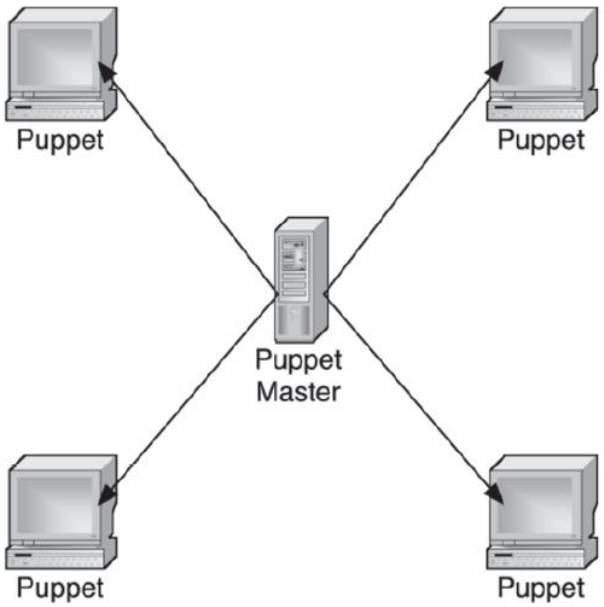
# 2 Puppet

Puppet on vapaan lähdekoodin ohjelmisto, jolla voidaan automatisoida järjestelmätasoon tehtäviä muutoksia keskitetysti sekä useammalla eri alustalla, käyttämällä puppetin omaa viitekehystä. Ohjelma on kirjoitettu Ruby – ohjelmointikielellä. (Puppet Labs 2012a & Puppet Labs 2012b.)

Puppet on yhteensopiva useiden ilmaisten, sekä kaupallisten käyttöjärjestelmien kanssa. Näitä esimerkiksi ovat Debian, Fedora, Mac OS X, Red Hat Enterprise Linux, Ubuntu, SuSE Linux Enterprise Server ja Microsoft Windows Server. (Puppet Labs 2012c.)

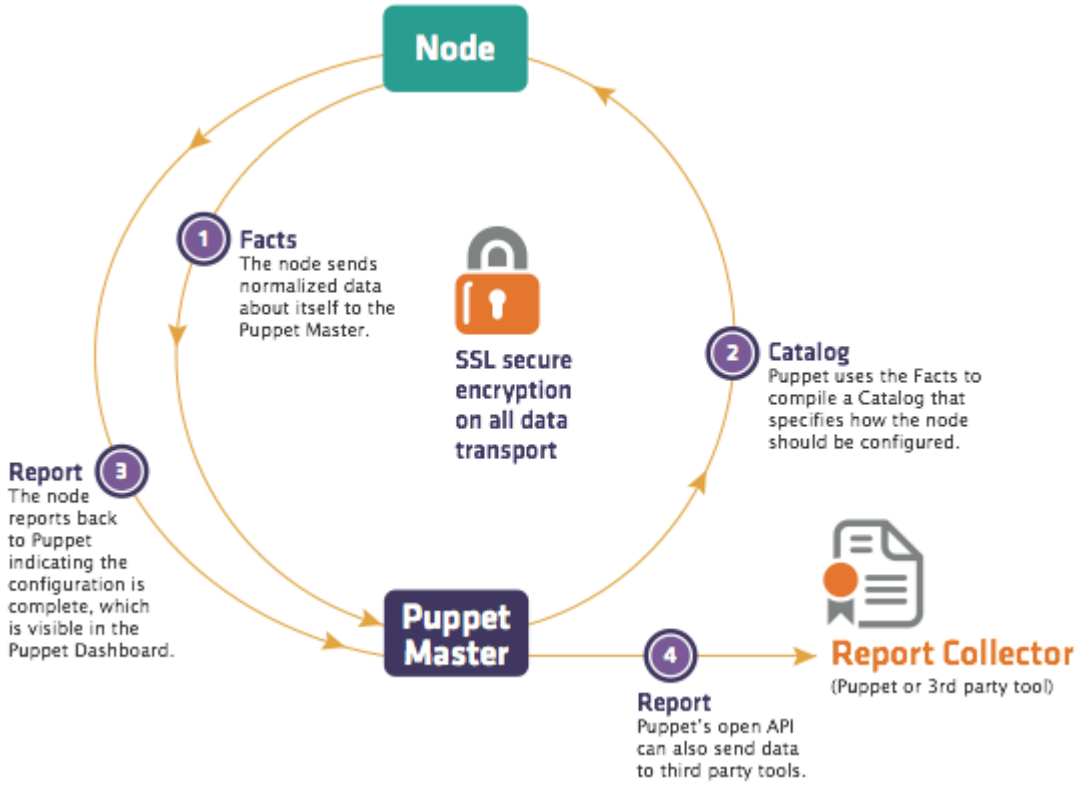
## 2.1 Toiminta

Puppet käyttää deklaratiivistä ohjelmointikieltä järjestelmäkonfigurointien muutosten ilmaisemiseen, jossa ongelma ratkaistaan esittämällä järjestelmälle haluttu ratkaisu tai tila. Tässä kontekstissa usein kerrotaan halutun ohjelmiston asennus tai konfiguroinnin muutos järjestelmälle. Levitysmallina käytetään ’client-server’ -tyyppistä ratkaisua, jossa palvelinkoneelle ilmaistaan haluttu muutos ja se levitetään päätekoneille. (Turnbull 2008, s.3-5.)




(Kuva 1. Puppet client-server –malli.)

Usein päätekoneiden Puppet ohjelmisto ajetaan taustaprosessiksi. Päätekone ajoittain lähettää palvelimelle tietoja olemassaolostaan ja kysyy katalogia haluttujen järjestelmäkomponenttien tilasta, ja kun päätekone vastaanottaa katalogin, niin se vertaa katalogissa ilmoitettujen resurssien tilan oman järjestelmänsä resurssien tiloihin, ja tarvittaessa muuttaa ne katalogin mukaisiksi. (Puppet Labs 2012e.)




(Kuva 2. Puppet client-server yhteysmalli.)

Järjestelmämuutokset puppetin läpi ovat idempotentteja, eli jos sama konfigurointi ajetaan useamman kerran järjestelmään, niin lopputulos on aina sama. Tästä suurena hyötynä on se, että järjestelmämuutokset voidaan ajaa useamman kerran järjestelmään turvallisesti. (Turnbull 2008, s.7.)

## 2.2 Manifestit, luokat ja resurssit

Manifestitiedostot ovat puppetin omia konfigurointitiedostoja, joiden tiedostopääte on aina ’.pp’. Manifestitiedostot voivat olla joko luokkatiedostoja, node-tiedostoja, tai näiden yhdistelmiä. Niitä voi linkata keskenään tai hajauttaa pienemminksi tiedostoiksi. (Puppet Labs 2012f.)

Manifestitiedostot usein erotellaan tiedostohierarkian perusteella, esimerkiksi node-tiedosto vakioasennuksessa on polussa ’/etc/puppet/manifests/site.pp’, ja tällä määritellään päätekoneet joihin halutut luokat levitetään. Luokkatiedosto usein rakennetaan erillisen moduulin alle, esimerkiksi ’/etc/puppet/modules/class_example/class.pp’, tätä kutsutaan nodetiedostosta, ja tämä tiedosto sisältää konkreettista logiikkaa päätekoneen muutoksia varten. (Puppet Labs 2012f && Puppet Labs 2012g.)

Resursseja käytetään ilmaisemaan haluttuja muutoksia luokkatiedostossa, jotka suoritetaan päätekoneen päässä. Mahdollista on korvata, asentaa, käynnistää, ja sammuttaa prosesseja tai palveluita, lisätä järjestelmälle ajastettuja ajoja, korvata tiedostoja tai ajaa keskitetysti komentoja. Laajalti kaikki järjestelmävalvojan tason muutokset on mahdollista tehdä puppetin resurssien kautta. (Puppet Labs 2012f && Puppet Labs 2012h.)

## 2.3 Tietoturva

Kaikki puppetin verkkoliikenne pohjautuu HTTPS suojatun yhteyden läpi tapahtuvaan liikenteeseen. Päätekoneen ja palvelinkoneen välillä on sertifikaattivarmennus, ja palvelinkoneen tulee erikseen hyväksyä päätekoneen sertifikaatti, jotta se suostuu kommunikoimaan päätekoneen kanssa. Jos päätekoneen sertifikaatti uusitaan varmennusprosessin jälkeen, niin se pitää varmentaa uudestaan palvelimella. (Puppet Labs 2012i.)

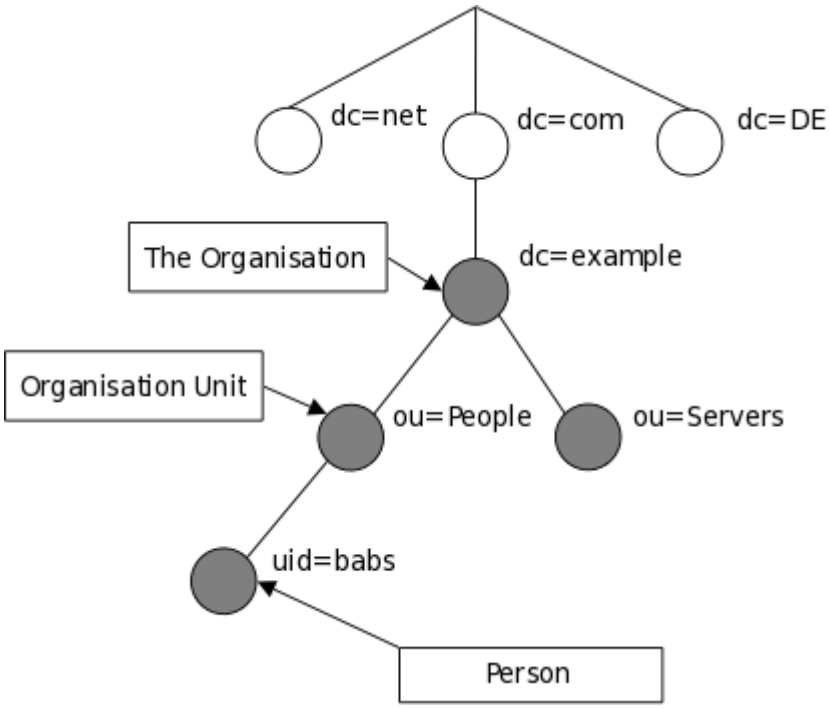
# 3 OpenLDAP

OpenLDAP on vuonna 1998 perustettu avoimen lähdekoodin hakemistopalvelu, joka on eriytynyt alkuperäisestä LDAP projektista. OpenLDAP kehitys jatkuu tänäkin päivänä. (OpenLDAP 2014.)

OpenLDAP käyttämä LDAP on hakemistopalveluiden (Active directory) käyttämä protokolla, joka pohjautuu X.500 hakemistopalveluiden malliin. LDAP –lyhenne tulee sanoista ”Light-weight Directory Access Protocol”. Tieto on järjestetty hakemistopuuhun avain-arvopapereina ja hakemistopuu rakentuu organisaation rakenteen ja/tai maantieteellisen sijainnin mukaan. LDAP protokollaa käyttää OpenLDAP lisäksi mm. Microsoft Active Directory ja FreeIPA. (OpenLDAP 2011a && Wikipedia 2016.)

## 3.1 Toiminta

LDAP hakemistopuun hierarkia rakentuu ylhäältä alaspäin, jos tietorakenne halutaan esittää graafisesti. Tieto alkaa latvasta, jossa voidaan määritellä joko organisaation maantieteellinen sijainti tai aloittaa latva suoraan organisaation rakenteesta. Ja mitä enemmän hakemistostopuussa liikutaan alaspäin, tiedosta tulee spesifisempää ja yksilöivämpää. Usein alimmilla tasoilla on käyttäjiä, päätekoneita tai oheislaitteita. (OpenLDAP 2011a.)




(Kuva 3. LDAP hakemistopuu.)

Tietoa käsitellään avain-arvopaperina, tarkoittaen että jos esimerkiksi haetaan tietoa käyttäjästä ’babs’ hakemistopuusta, niin tieto haettaisiin

”uid=babs,ou=People,o=example,dc=com” tunnisteella. Käytännössä tieto aina haetaan mainitsemalla täydellinen polku kohteeseen, tässä tilanteessa käyttäjätietoon. Tästä palautuu arvopaperi, joka sisältää tarkentavia tietoja käyttäjästä, toisin sanoen erikseen määritettyjä attribuutteja, ja näihin tietoihin verrataan järjestelmätasolla erilaisia toiminnallisuuksia, kuten autentikointia tai haetaan yleisemmin informatiivista tietoa, kuten käyttäjän todellista nimeä. (OpenLDAP 2011a.)

### 3.2 Tietoturva

LDAP palvelimelle usein muodostetaan yhteys yksinkertaisella autentikoinnilla, käytännössä tämä tarkoittaa sitä, että jos otetaan yhteys LDAP palvelimelle ja haetaan tietoa käyttäjästä, niin palvelin palauttaa sen automaattisesti. Etuna tässä on LDAP palvelimen läpinäkyvyys, eikä käyttäjä tiedä keskitetystä järjestelmästä mitään. Ongelmaksi tässä koituu se, että kuka tahansa käyttäjä, joka palvelimelle saa yhteyden voi nähdä koko LDAP hakemistopuun ja sen tiedot. Tähän on ratkaisuna tehty tuki sertifikaattien käytölle LDAP palvelimen ja päätekoneen välisessä yhteyden muodostuksessa. Jos päätekone haluaa ottaa yhteyden LDAP palvelimelle, niin sillä tulee olla palvelimen allekirjoittama sertifikaatti, muutoin palvelin ei vastaa päätekoneen kutsuihin. (OpenLDAP 2011b.)

Avain-arvopaperien arkaluontoisten tietojen, kuten salasanan, suojaus on tärkeää. Ja vakiona LDAPin schema-rakenteessa käyttäjän salasana on selkokielisenä, tätä on perusteltu sillä, että voidaan suoraan tukea useampaa salausmenetelmää kuten DIGEST-MD5 tai SHA-suojauksia. Suositus kumminkin on se, että arkaluontoinen tieto olisi aina suojattuna myös LDAPin kantatasolla. Käyttämällä LDAPin SSHA schemaa, voidaan käyttäjän salasana suojata kantaan asti. Kantatasolla, käyttäjällä voi olla useampia eri salasanoja, eli jos ongelmaksi tulee se, että eri järjestelmät käyttävät eri suojauksia, niin pystytään usea eri salasana erillisellä suojauksella linkittää käyttäjään.
(OpenLDAP 2011b.)

# 4 SSSD – System Security Services Daemon

Nykypäivänä autentikaatio tapahtuu usein keskitetyn autentikointipalvelimien läpi, kuten LDAP tai Kerberos, ja ennen tällainen konfigurointi Linux koneilla tarvitsi usein useita eri komponentteja ja konfigurointeja toimiakseen. Tästä seurasi ongelmia ylläpidettävyydessä. Toisekseen ongelmaksi koitui myös käyttäjätietojen varastointi, vaikka komponentteja olikin tehty LDAP ja kerberos yhteyksiä varten, niin näissä ei itsessään ollut vielä linjattomalle käytölle tukea, joten käyttäjätietojen varastointia varten päätekoneelle tuli vielä erikseen tehdä omia konfigurointeja. SSSD on ”all-in-one” komponentti, jonka tarkoituksena on tuoda nämä erilliset komponentit yhden komponentin kautta hallittavaksi. (LWN.net 2011.)

SSSD on useissa UNIX- järjestelmissä käytetty järjestelmän taustaprosessi, ja sen tarkoituksena on mahdollistaa käyttäjän identifiointi ja autentikointi yhden viitekehyksen läpi. SSSD tarjoaa käyttäjän tietojen varastoinnin ja linjattoman käytön, sekä PAM ja NSS moduulien suoran konfiguroinnin. (Fedorahosted 2016a.)

SSSD mahdollistaa autentikoinnin OpenLDAP, Kerberos, Redhat FreeIPA, Microsoft Active directoryn, sekä Samba4 active directoryn kautta. Ominaisuutena myös on, että SSSD vähentää rasitusta vanhan mallin ”nss_ldap” identifikaatio malliin verrattuna, jossa jokainen eri ohjelma järjestelmästä kysyy suoraan identifikaatiopalvelimelta käyttäjätietoja. SSSD mallissa, vain itse SSSD prosessi keskustelee autentikaatio- ja identifikaatiopalvelimen kanssa. (Fedoraproject 2010 & Ubuntu Wiki 2014.)

Useamman erillisen toiminimen (domain) samanaikainen käyttö on myös mahdollistettu. Tämä tarkoittaa sitä, että päätekoneelle voidaan konfiguroida useampi LDAP palvelin tai muu autentikointi- tai identifointipalvelin toimimaan rinnakkain, vaikka ne ajaisikin aivan samaa asiaa. (Fedoraproject 2010.)

# 5 Keskitetyn ohjelmistohallinnan toteutusosuus

Tässä kappaleessa käsitellään toteutusosuuden eri vaiheita, ja konkreettisia muutoksia joita tehdään keskitetyn ohjelmistonhallinnan kautta päätekoneille, jotta LDAP saadaan toimimaan autentikointimenetelmänä. Tilanne on siis fiktiivinen, jossa yrityksellä on käytössä LDAP, mutta kaikki LDAP autentikointiin liittyvät konfiguraatiotyöt on tehty käsin aina uuden, huolletun tai vaihdetun työpisteen käyttöönoton yhteydessä.

## 5.1 OpenLDAP

Yrityksellä on käytössä openLDAP keskitetty käyttäjänhallinta, jota päätekoneet käyttävät. Tässä kappaleessa käydään läpi LDAP palvelun manuaalinen käyttöönottoprosessi päätekoneella, ja selvitetään mistä tähän tarvittavat tiedot saadaan haettua.

### 5.1.1 Käyttötapaukset

- Authentikointi päätekoneille (SLED12) tulee tapahtua LDAP palvelun kautta.
- Jos LDAP palvelin on kaatunut tai siihen ei saada yhteyttä, tulee käyttäjällä, joka on ainakin kerran onnistuneesti kirjaunut sisään järjestelmään, pystyä kirjautumaan omalla tunnuksellaan järjestelmään.
- Päätekoneiden (SLED12) SUDO oikeudet ylläpidetään LDAP palvelun kautta, ei päätekoneilla.
- Päätekoneille ei ole perustettu paikallisia käyttäjiä (pl. asennuksen mukana vaadittu paikallinen pääkäyttäjätunnus)
- Koneet käyttävät LDAP yhteyttä ilman TLS suojausta, koska kyseessä on yrityksen oma suojattu sisäverkko.

### 5.1.2 Selvitysosuus – palvelin

SLES 12 palvelimelta selvitettiin LDAP palvelimen rakenne. Alla yhteenveto tärkeimmistä asioista, jotka selvittelyosuudessa tulivat ilmi:

- Palvelimen osoite: 10.0.2.15
    - Tieto löytyy /etc/hosts
- Palvelimen nimi (hostname): puppet
    - Tieto löytyy ajamalla komento ’uname –n’
- Palvelimen FQDN: puppet.com
    - Tieto löytyy joko DNS palvelimelta tai /etc/hosts
- LDAP schema: rfc2307bis
    - YaST2 -> Authentication server

- LDAP BaseDN: dc=puppet,dc=com
    - Ajamalla komento ’ldapsearch –x’
- LDAP SudoBaseDN: ou=SUDOers,dc=puppet,dc=com
    - Ajamalla komento ’ldapsearch –x’
- LDAP yhteysosoite: ldap://puppet.com
    - Tieto löytyy /etc/ldap.conf tai /etc/openldap/ldap.conf

Yllämainitut tiedot riittävät LDAP autentikoinnin konfigurointiin päätekoneelta.

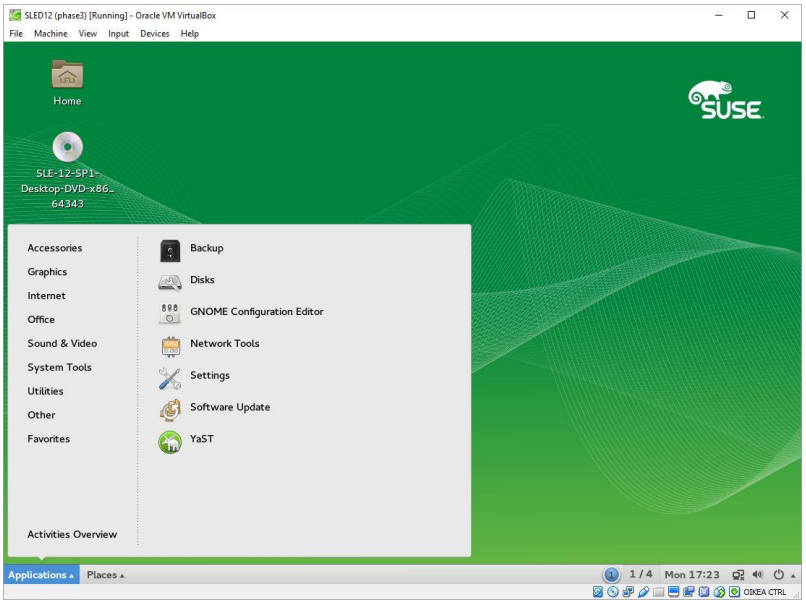
### 5.1.3 Selvitysosuus - Päätekone

Selvitystyö alkoi sillä, että konfiguroitiin manuaalisesti LDAP autentikointia varten asetukset päätekoneelle, ja selvitettiin, että mitä kaikkia muutoksia SLED12 omat työkalut (Yast2) tekivät järjestelmään. Alla selitetty kuvien avulla konfigurointiprosessi LDAP autentikoinnin käyttöönotosta käyttöjärjestelmän omien työkalujen avulla.

Juuri ennen itse käyttöönotto prosessin aloittamista, muodostettiin lista tämänhetkisistä paketeista päätekoneelle, polkuun ’/tmp/package.list’. Tästä saadaan aikaleima ja tarkka pakettilistaus tilanteesta, ennen kuin LDAP autentikaatio on otettu käyttöön. Kappaleen 6.1.3 lopussa selitetään tarkemmin tämän tiedoston tarkoitus.

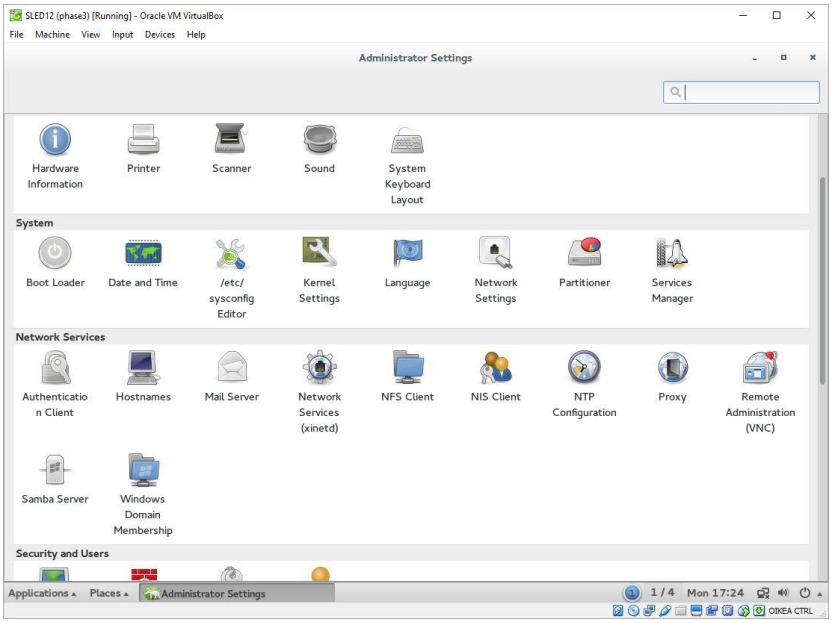
LDAP autentikoinnin konfigurointi

1. Avataan Applications valikosta YaST ja syötetään pääkäyttäjän salasana



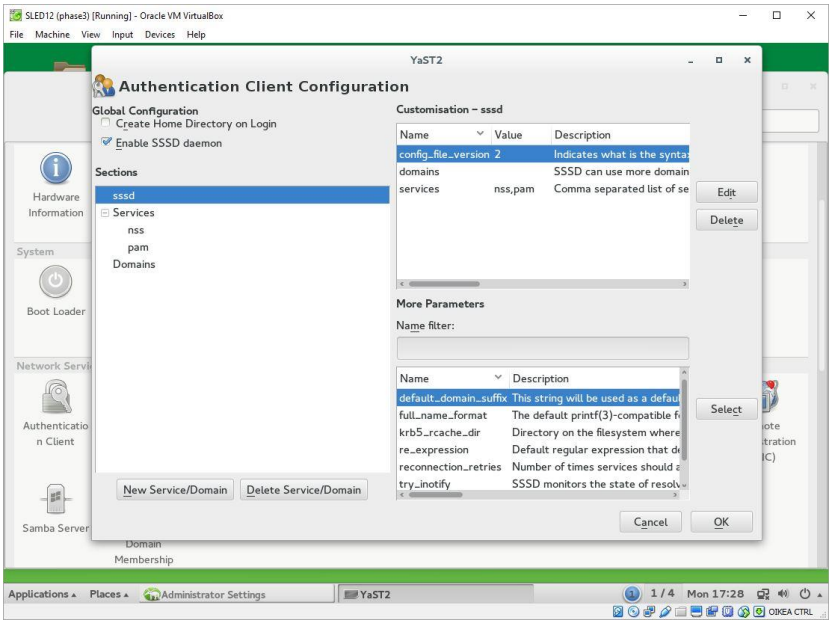

(Kuva 4. Kohta 1.)

2. Valitaan ’Authentication Client’, ’Network Services’ –välilehden alta




(Kuva 5. Kohta 2.)

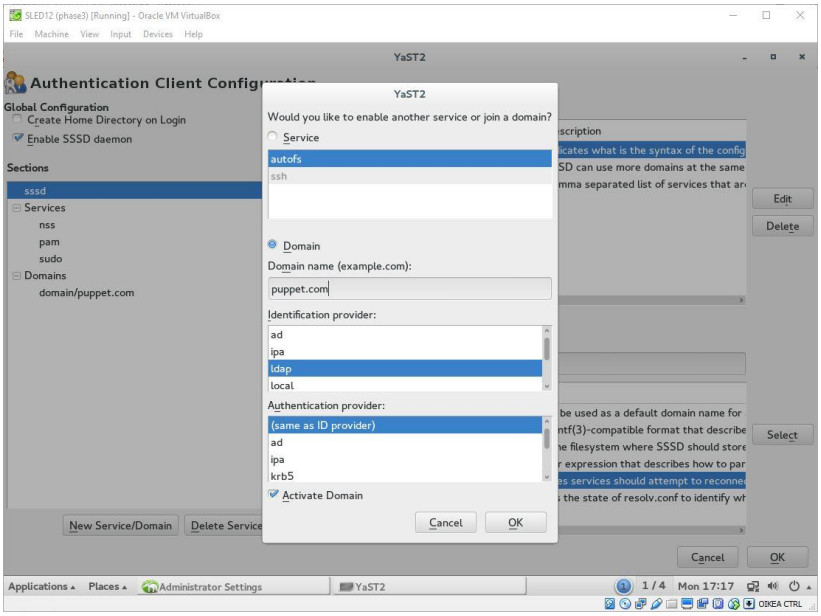
3. Lisätään uusi domain, painamalla New Service/Domain




(Kuva 6. Kohta 3.)

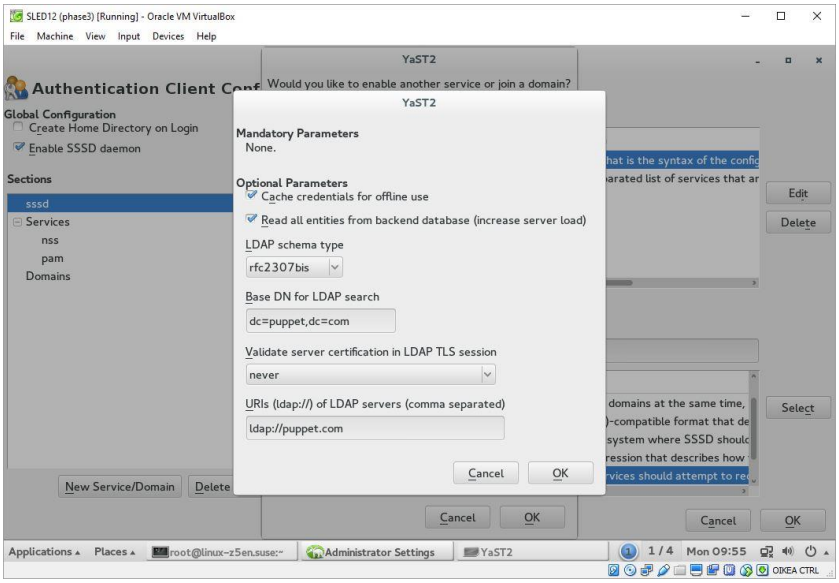
4. Ensin syötetään, että onko kyseessä palvelu (nss,pam,sudo) vai palvelin (domain). Valitaan ’Domain’ ja ilmoitetaan tietoja 2.1.2 kohdasta löydettyjen tietojen mukaan. Identifikaation tarjoaja (Identification provider) –kenttään

ilmoitetaan ’ldap’, ja Autentikaation tarjoaja (Authentication provider) – kenttään ilmoitetaan ’(same as ID provider)’. Sivun asetukset kuitataan painamalla ’OK’.



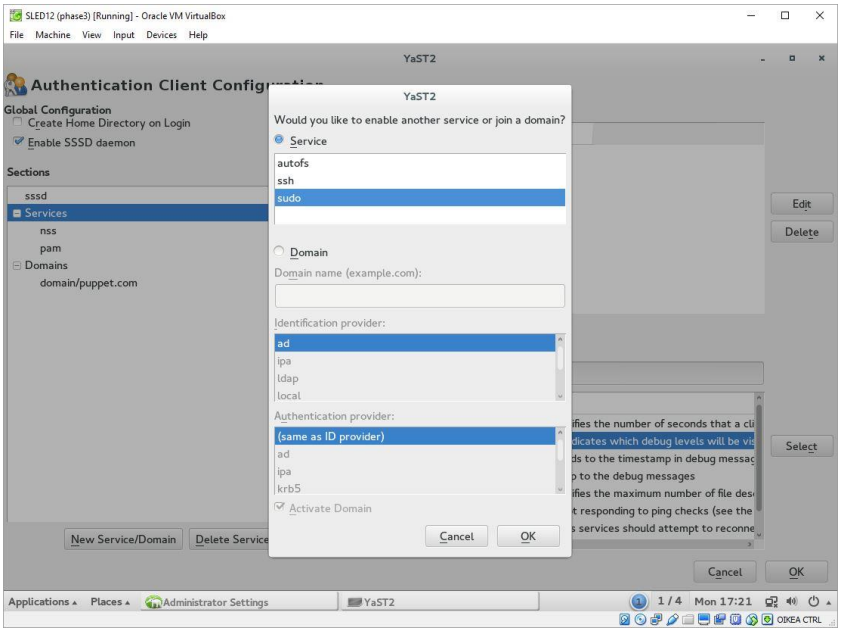

(Kuva 7. Kohta 4.)

5. Optional Parameters -kohdasta valitaan ’Cache credentials for offline use’ , jolla SSSD hakee kirjautumisen yhteydessä LDAP käyttäjän tiedot koneen omaan välimuistiin. LDAP mallin (schema) tyyppinä on sama kuin 2.1.2 kohdan ’rfc2307bis.’ ’Base DN for LDAP search’ on sama kuin 2.1.2 LDAP BaseDN. Tämä kertoo, mistä lähdetään tietoa hakemaan LDAP schemasta. ’ Validate Server certification LDAP TLS session ’never’. Tämä tarkoittaa sitä, että muodostetaanko LDAP yhteys käyttäen sertifikaatteja palvelimelle. ’URIs (ldap://) of LDAP servers (comma separated)’ ilmoitetaan LDAP palvelimen (tai palvelimien) osoitteet, joista autentikointitietoja haetaan. Kuitataan sivun asetukset painamalla ’OK’



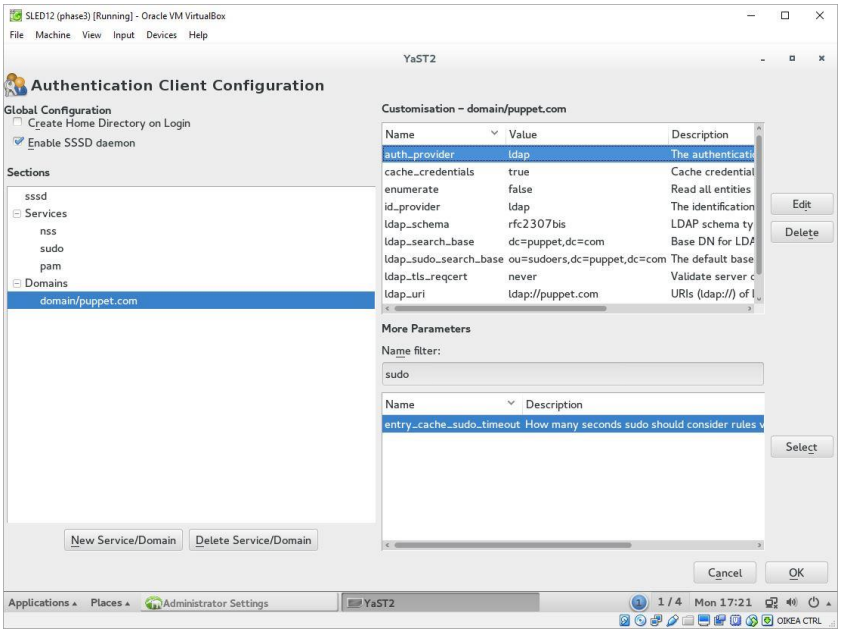

(Kuva 8. Kohta 5.)

6. Lisätään SUDO tuki SSSD:lle, joka ohjaa, että kuka saa ajaa ja mitä komentoja saa ajaa päätekoneelta järjestelmänvalvojan oikeuksilla. Ensin valitaan ’New Service/Domain’. Valitaan tyypiksi ’Service’ ja alla olevasta listasta ’sudo’ ja kuitataan OK.




(Kuva 9. Kohta 6.)

7. Valitaan juuri luotu ’domain/puppet.com’ vasemman puoleisesta listasta. Ruudun oikealla puolella näkyvästä ’More Parameters’ kohdasta haetaan avainsanalla ”ldap_sudo_search_base” ja valitaan ruudun oikealta laidalta ’Select’ ja lisätään arvoksi ’ou=SUDOers,dc=puppet,dc=com’ ja kuitataan OK.



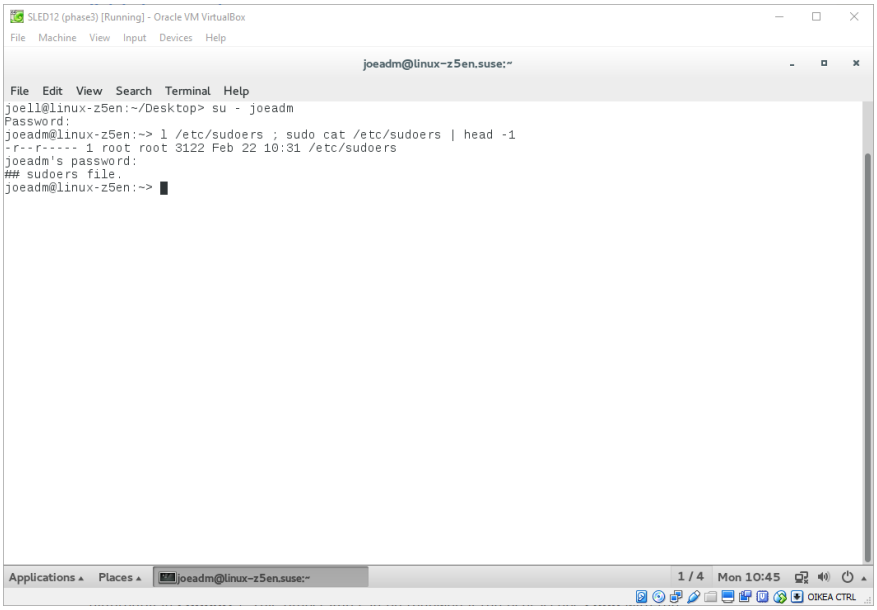

(Kuva 10. Kohta 7.)

8. Samalla tavalla kuin kohdassa 7. haetaan avainsanalla ’sudo_provider’, valitaan ’Select’, ilmoitetaan arvoksi ’ldap’ ja kuitataan OK. Tämä kertoo, että SUDO käyttäjäoikeuslistaukset tulevat myös LDAPilta.

9. Kaikki muutokset kuitataan painamalla ’OK’. Tämän jälkeen järjestelmä asentaa tarvittavat paketit ja asettaa juuri määritetyt konfiguraatiot. Pakettiasennuksien ja määritysten jälkeen on vielä kerran käynnistettävä ’Authentication client’ ja kuitattava se kiinni ’OK’, koska ohjelma jättää ensimmäisellä kerralla määrittämättä PAM autentikaatiomoduuleihin tärkeitä määrityksiä liittyen SSSD.

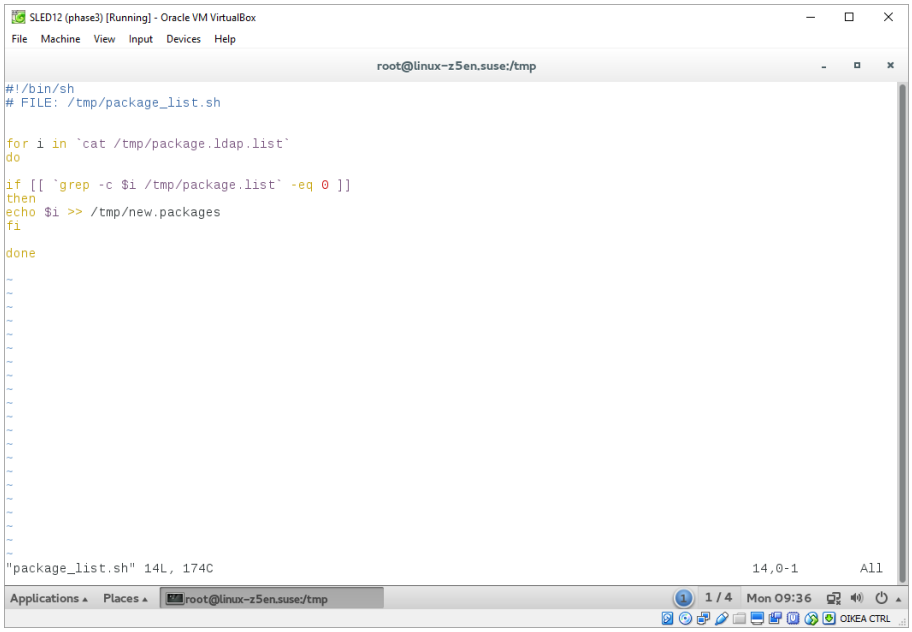
10. Viimeisenä on kommentoitava seuraavat 2 riviä pois /etc/sudoers tiedostosta, jotta sudo komennot toimivat.
    a. Defaults targetpw # Ask for the password of the target user i.e. root
    b. ALL ALL=(ALL) ALL #WARNING! Only use this together with ’Defaults targetpw’!

11. Testataan LDAP tunnuksen ja SUDOn toimivuus. Kirjaudutaan sisään LDAP tunnuksella ja ajetaan komento ’sudo’ –etuliitteellä.



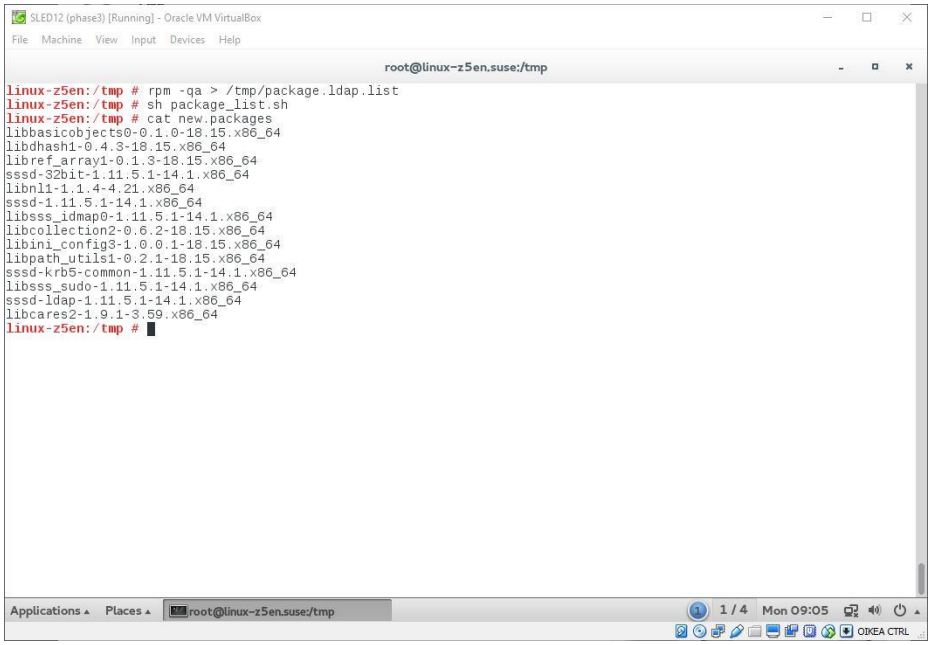

(Kuva 11. Kohta 11.)

Toimivan käyttöönoton jälkeen, haettiin kaikki muutokset järjestelmätasolta, jotka ovat muuttuneet käyttöönoton yhteydessä. Ensin haetiin kaikki uudet asennetut paketit järjestelmästä. Tehtiin skripti, jolla verrattiin kahden listatiedoston sisältöä ja tulostettiin vain puuttuvat rivit uuteen tiedostoon (Kuva 12.). Tässä tilanteessa verrataan pakettilistaa, joka muodostettiin juuri ennen asennusta (/tmp/package.list) ja uutta pakettilistaa, joka muodostettiin käyttöönoton jälkeen (/tmp/package.ldap.list).




(Kuva 12. Pakettilistauksen vertailu skripti.)

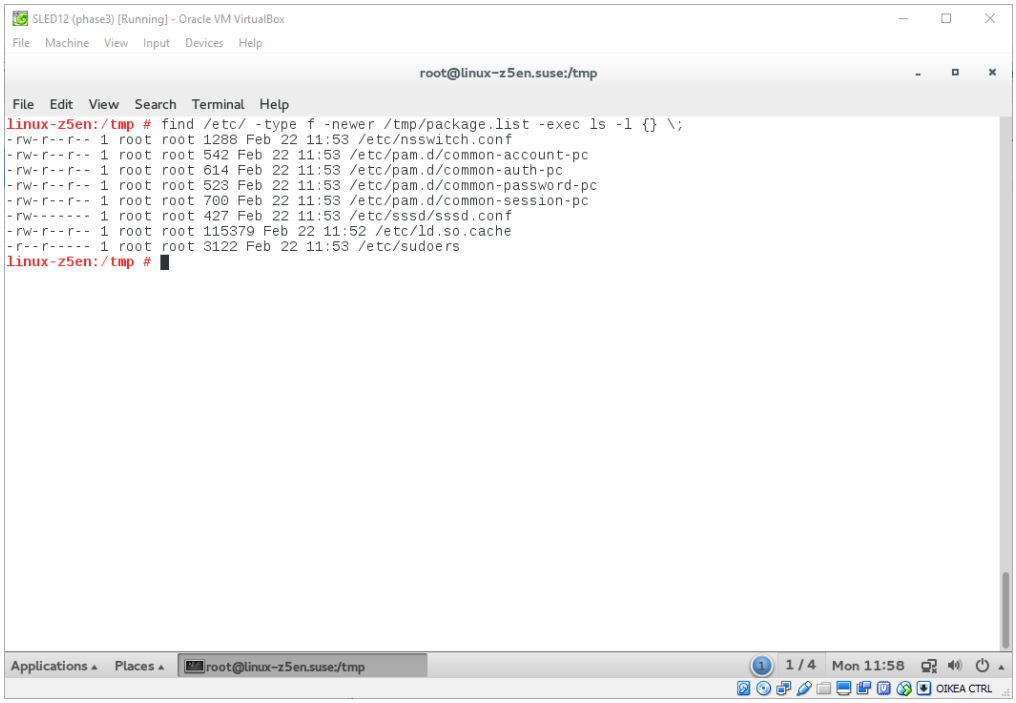
Tämän jälkeen tehtiin uusi pakettilistaus järjestelmätasolla käyttöönoton jälkeen, ajetiin skripti ja listattiin uudet paketit, jotka olivat asentuneet käyttöönoton jälkeen (Kuva 13.)




(Kuva 13. Pakettilistauksen vertailuskriptin tulostiedoston sisältö.)

Sitten haettiin kaikki konfiguraatiotiedostot, joihin on tehty muutoksia käyttöönoton yhteydessä.

Tämä onnistui käyttämällä juuri ennen asennusta tehtyä listaa asennetuista paketeista (/tmp/package.list) ja hakemalla järjestelmästä kaikki sitä uudemmalla aikaleimalla (timestamp) olevat tiedostot /etc/ -hakemistosta (Kuva 14.). Komennot suoritettiin pääkäyttäjänä (root).




(Kuva 14. Aikaleima vertailu järjestelmätason muuttuneista tiedostoista.)

Listuksista voitiin päätellä, että käyttöönotto on vaikuttanut seuraaviin konfigurointeihin:

- PAM konfiguraatiotiedostoihin.
- sssd.conf- konfiguraatiotiedostoon.
- nsswitch.conf –konfiguraatiotiedostoon.
- sudoers - konfiguraatiotiedostoon

Nämä olivat oleellisimmat tiedot, joita tarvittiin LDAP autentikoinnin käyttöönottoon SLED12 päätekoneella.

### 5.2 Keskitetty ohjelmistonhallinta puppet

Kun manuaalisen LDAP autentikaation käyttöönoton prosessi päätekoneella on selvä, voidaan alkaa keskittymään itse keskitetyn LDAP käyttöönoton rakentamiseen. Ennen tätä vaihetta on tärkeä olla selvillä seuraavat asiat:

- Mitä paketteja puppetin tulee asentaa?
- Mitä konfiguraatiotiedostoja puppetin tulee korvata?
- Mitkä konfiguraatiotiedostojen lukuoikeudet ovat?
- Mitä palveluita (services) tulee käynnistää uudelleen tiedostokorvausten jälkeen, jotta uudet konfiguroinnit tulevat voimaan?

Seuraavat paketit, sekä niiden riippuvuudet (dependencies) tulee asentaa:

- sssd
- sssd-ldap

Seuraavat konfiguraatiotiedostot tulee korvata, ja asettaa niille seuraavat tiedostooikeudet:

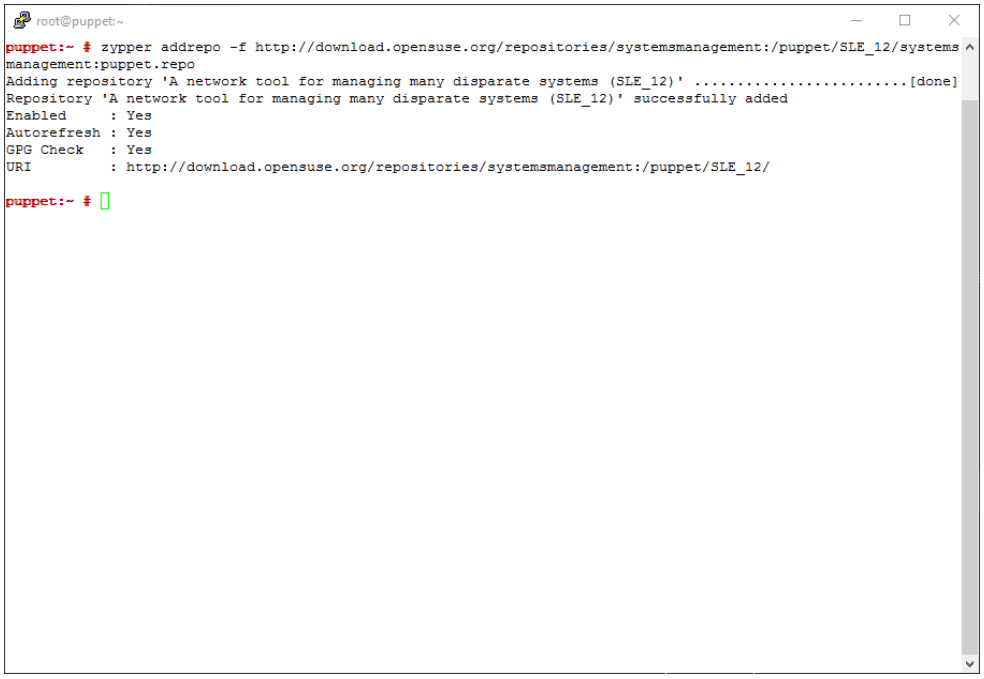
| | | |
|---|---|---|
| • /etc/pam.d/common-auth-pc | 644 | root:root |
| • /etc/pam.d/common-account-pc | 644 | root:root |
| • /etc/pam.d/common-password-pc | 644 | root:root |
| • /etc/pam.d/common-session-pc | 644 | root:root |
| • /etc/nsswitch.conf | 644 | root:root |
| • /etc/sssd/sssd.conf | 600 | root:root |
| • /etc/sudoers | 600 | root:root |

Seuraavat palvelut tulee käynnistää uudelleen konfiguraatiotiedostojen korvauksen jälkeen:

- sssd

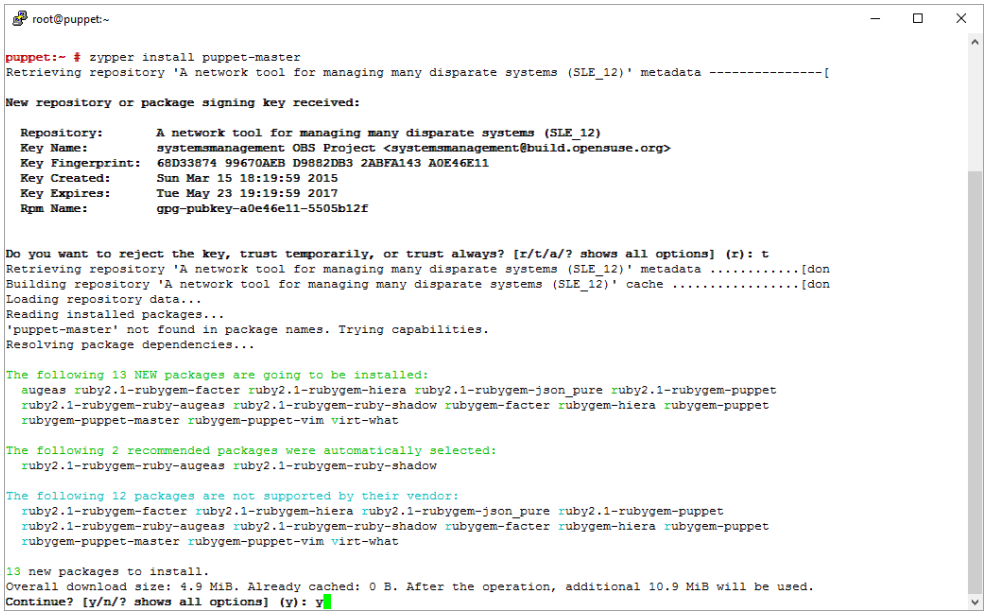
### 5.2.1 Puppet-master asennus ja testaus SLES 12 palvelimella

Koska SuSEn omista repositorioista ei löydy virallista versiota puppetille, joten on lisättävä erillinen repositorio, josta haetaan viimeisin vakaa versio. joka on käännetty SLE 12 (Kuva 15.). Komennot ajetaan root –käyttäjänä.




(Kuva 15. Repositorion lisäys palvelinkoneella.)

Asennetaan puppet-master, ja koska lisättiin erillinen repositorio, josta käyttöjärjestelmällä ei ole tietoa, tulee ensin varmistaa repositoriosta saatu varmenne oikeaksi (Kuva 16.)




(Kuva 16. Repositorion varmennus palvelinkoneella.)

Asennuksen yhteydessä puppet-master generoi itsellensä sertifikaatin, jota vasten varmistetaan aina uudet päätekoneet (client) sitä mukaan kun ne halutaan mukaan puppetin toiminta-alueen piiriin.

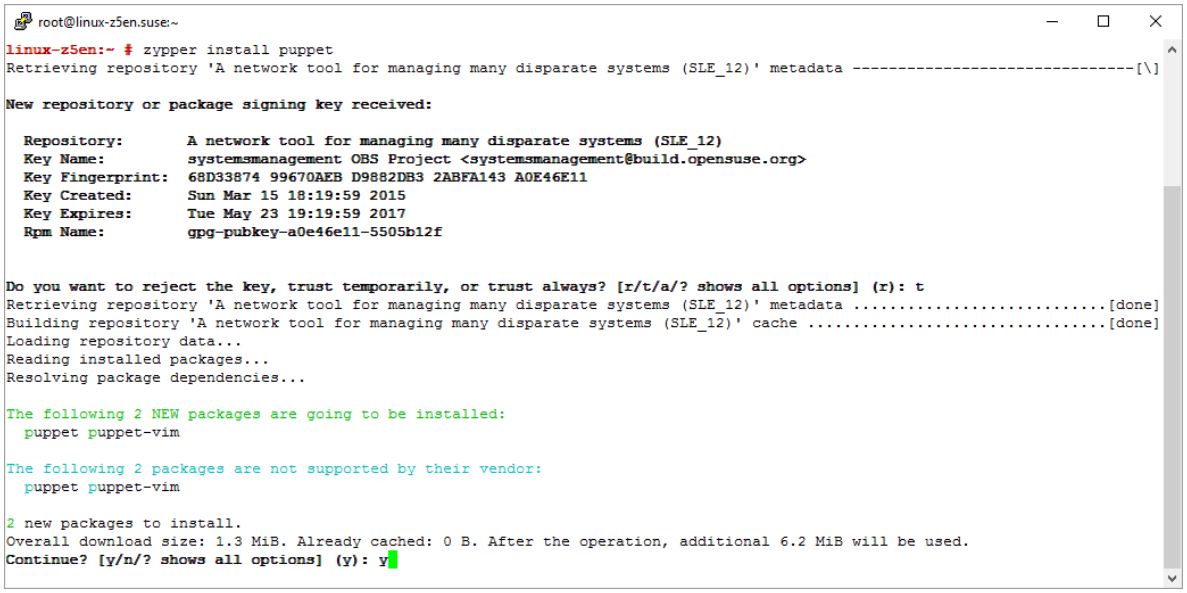
Puppet-master käynnistyksen jälkeen kuuntelee porttia 8140, ja koska koko projekti tapahtuu sisäverkon sisällä, ei tarvinnut käyttöjärjestelmän palomuurille erikseen määritellä kyseistä porttia auki.

Puppet-masterin yleistä toimivuutta voidaan testata määrittämällä yksinkertainen manifesti ja ajamalla se paikallisesti. Luodaan manifesti joka korvaa ’/etc/motd’ tiedoston muutetulla ’/etc/puppet/files/motd’ –tiedostolla, ja asettaa tiedostolle omistajaksi ja omistajaryhmäksi ’root’, sekä lukuoikeuksiksi 644 (-rw-r—r—). (Kuva 17.)


(Kuva 17. Yksinkertainen puppet manifesti testi.)

### 5.2.2 Puppet asennus ja testaus SLED 12 päätekoneella


(Kuva 18. Repositorion lisäys päätekoneella.)

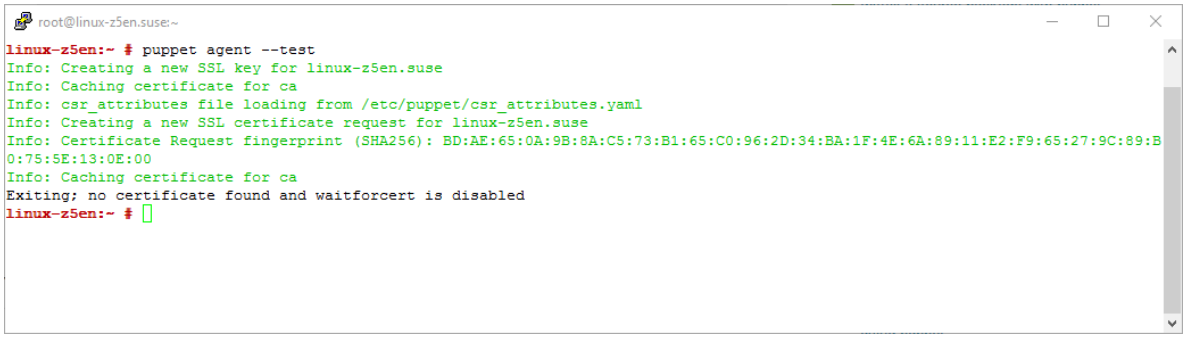



(Kuva 19. Repositorion varmennus päätekoneella.)

Vakiona puppet-client hakee joko DNS palvelimelta tai /etc/hosts –tiedostosta palvelimesta tietoa nimellä ’puppet’. Tämän tiedon pystyy muuttamaan /etc/puppet/puppet.conf –tiedostosta. ’server’ –kenttään ilmoitettu parametri tulee täsmätä palvelimen nimen (hostname) kanssa, muutoin SSL varmennus epäonnistuu. (Kuva 20.)


(Kuva 20. /etc/puppet/puppet.conf –sisältö.)

Tehdään testiyhteys puppet palvelimelle päätekoneelta, samalla tämä muodostaa palvelimelle sertifikaatin hyväksymispyynnön, jonka palvelin pitää ensin hyväksyä, jotta puppet muutoksia voidaan alkaa levittämään ko. päätekoneelle. Parametri '--test' tarkoittaa sitä, että testataan vain yhteys puppet-master palvelimelle, eikä haeta muutoksia palvelimelta. (Kuva 21.)




(Kuva 21. Päätekoneen Puppet testiyhteys palvelimelle.)

Siirrytään palvelinkoneelle, listataan odottavassa tilassa (pending) olevat uudet sertifikaatit, ja varmistetaan päätekoneen sertifikaatti oikeaksi. (Kuva 22.)


(Kuva 22. Palvelinkoneen Puppet sertifikaattivahvistus päätekoneelle.)

Palvelinkoneella, lisätään aikaisemmin tehtyyn manifestiin keltaisella näytetyt kohdat, ja ajetaan muutos levitettäväksi. 'node' kertoo kohteen, jonne kyseinen manifesti halutaan levittää. Arvo 'default' levittää sen kaikille päätekoneille, jota puppetmasterilla on tiedossa ml. myös itseensä. (Kuva 23.)


(Kuva 23. Manifestin muokkaus päätekonetta varten.)

Päätekoneella, haetaan aikaisemmin tehdyn yksinkertaisen manifestin (/etc/motd) muutokset päätekoneelle. Parametrit ’—onetime’ ja ’—no-daemonize’, tarkoittavat että haetaan vain muutokset puppet-masterilta ja suljetaan puppet. (Kuva 24.) Ilman näitä parametrejä puppet jäisi taustaprosessiksi odottamaan uusia muutoksia palvelimelta.


(Kuva 24. Manifestin haku päätekoneelle palvelimelta.)

Päätekoneelle siis yhteys toimii, koska palvelinkoneen manifestiin määritetty muutos toteutui myös päätekoneella.

### 5.2.3 Arkkitehtuuri

Tässä alakappaleessa käydään läpi projektin puppet-master manifestien arkkitehtuuri.

Lopullisessa toteutuksessa käytetään yhtä manifestia, joka kutsuu kolmea eri luokkaa (class), jotka ajetaan tietyssä järjestyksessä. Levitysprosessi alkaa puuttuvien palveluiden asennuksella, sitten konfiguraatiotiedostojen korvauksella, ja viimeisenä palveluiden konfiguroinneilla sekä palveluiden uudelleenkäynnistyksillä.

Tiedostohierarkia projektissa on seuraavanlainen:
*Puppet manifest-, moduuli- ja luokkatiedostot*

*/etc/puppet/manifests/site.pp*
*/etc/puppet/modules/sled12ldap/manifests/init.pp*
*/etc/puppet/modules/sled12ldap/manifests/ldap_configuration.pp*
*/etc/puppet/modules/sled12ldap/manifests/ldap_packages.pp*
*/etc/puppet/modules/sled12ldap/manifests/ldap_services.pp*

*Konfiguraatiotiedostot*

*/etc/puppet/modules/sled12ldap/files/sudoers*
*/etc/puppet/modules/sled12ldap/files/sssd/sssd.conf*
*/etc/puppet/modules/sled12ldap/files/pam/common-account*
*/etc/puppet/modules/sled12ldap/files/pam/common-auth*
*/etc/puppet/modules/sled12ldap/files/pam/common-password*
*/etc/puppet/modules/sled12ldap/files/pam/common-session*

Konfiguraatiotiedostot (/etc/puppet/modules/sled12ldap/files/) ovat haettu palvelimelle suoraan päätekoneelta (SLED 12), jossa on otettu käyttöön LDAP autentikointi. Nämä eivät ole yksilöiviä tietoja päätekoneesta, vaan LDAP autentikoinnin yhteyden konfigurointitiedostot, jotka voidaan vapaasti levittää (ainakin tässä tilanteessa) muihinkin SLED12 päätekoneisiin.

Kun konfiguraatiotiedostot on viety palvelimelle, on varmistettava, että tiedostoilla ja hakemistoilla on sellaiset tiedosto-oikeudet, että puppet-master -palvelun käyttämä ’puppet’ -käyttäjä pystyy lukemaan niitä.

Manifestin kutsuminen alkaa ’*/etc/puppet/manifests/site.pp*’ tiedostosta. Tiedosto sisältää tiedon kutsuttavista moduuleista ja siitä mihin kutsuttava moduuli levitetään. Tässä projektissa kutsutaan moduulia ’sled12ldap’ ja se jaetaan vain koneelle ’linux-z5en.suse’. (Liite 1.)

Seuraava vaihe on moduulin luokkakutsutiedosto ’*/etc/puppet/modules/sled12ldap/manifests/init.pp*’, jossa määritellään 3 eri luokkaa joita kutsutaan. Ensin kutsutaan ’ldap_packages’ -, toisena ’ldap_configuration’ - ja viimeisenä ’ldap_services’ -luokkaa.
(Liite 2.)

Moduulin käyttämät luokat ovat erillisiä luokkatiedostoja moduulin kotihakemistossa (*/etc/puppet/modules/sled12ldap/manifests/*), jotka sisältävät päätekoneelle tehtävät muutokset. Vakiona puppet-master hakee moduuleita oman kotihakemiston modules –hakemistosta.

*’/etc/puppet/modules/sled12ldap/manifests/ldap_packages.pp’* - Luokka asentaa selvitysvaiheessa todetut tarvittavat paketit järjestelmään.
*(Liite 3.)*

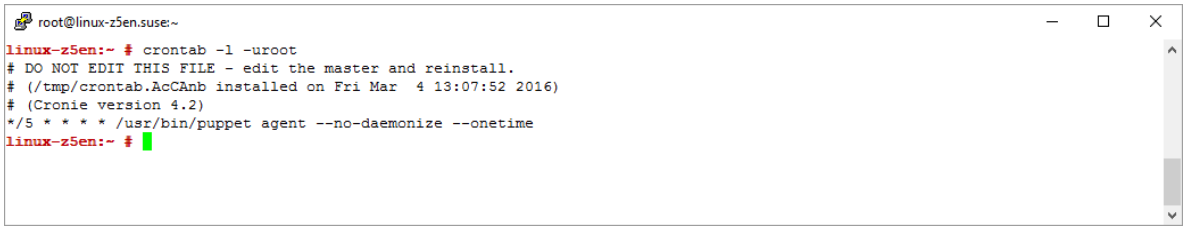
’*/etc/puppet/modules/sled12ldap/manifests/ldap_configuration.pp*’ – Luokka korvaa selvitysvaiheessa todetut konfiguraatiotiedostot päätekoneella ja asettaa niille tiedosto-oikeudet.
(Liite 4.)

’*/etc/puppet/modules/sled12ldap/manifests/ldap_services.pp*’ – Luokka asettaa ’sssd’ –palvelun käynnistymään aina koneen käynnistyksen yhteydessä ja pakottaa sen käynnistyksen kun manifesti ajetaan. ’subscribe’ – parametri on riippuvuusparametri, siihen kun sssd.conf –tiedosto korvataan, niin ’sssd’ -palvelu käynnistetään uudelleen.
(Liite 5.)

Tämän projektin liitteinä ovat kaikki konfiguraatiotiedostot, joita projektissa on käytetty.

### 5.2.4 Testaus ja virheiden selvittelyt

Testaus suoritettiin siten, että asennettiin kohdan 4.2.2 mukaisesti puppet päätekoneelle, ja tehtiin ajastettu komento (cron), joka tarkastaa 5 minuutin välein muutokset puppet-masterilta, siten ettei se jätä taustaprosessia odottamaan muutoksia puppet-masterilta. (Kuva 25.)

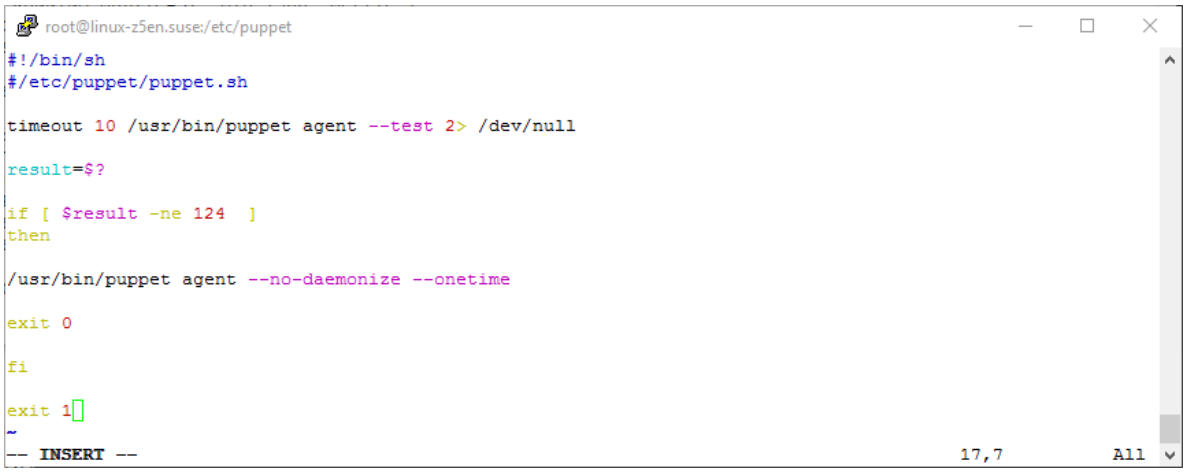



(Kuva 25. Puppet komento lisättynä ajastettuihin ajoihin.)

Virheenkäsittelyä ei ole äärettömän hyvin hoidettu puppet agentin päässä, ja yhdessä tilanteessa huomattiin, että jos SSL suojauden varmennus epäonnistuu, niin puppet jää taustalle kummittelemaan, tekemättä mitään. Ongelmaksi tässä tulee kaksi asiaa:

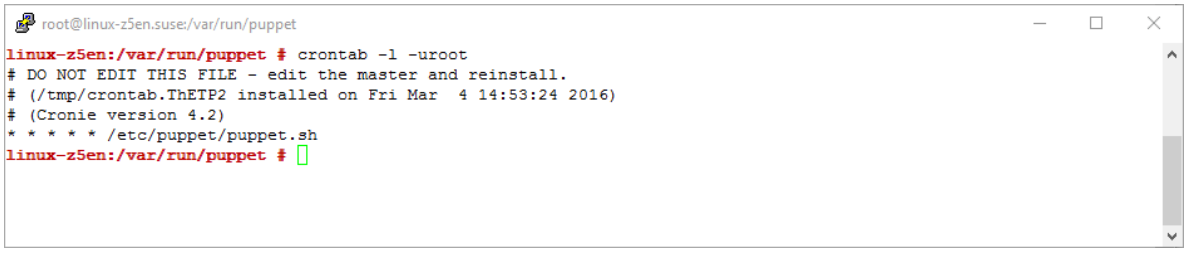
1. Uutta puppet prosessia ei käynnisty, jos ohjelma huomaa, että puppet on jo käynnissä.
2. Kummitteleva prosessi ei sammu kuin sulkemalla puppet prosessi tai käynnistämällä kone uudestaan.

Tämän takia kirjoitettiin skripti, joka varmistaa sen, ettei yhteys jää jumiin. Jos testiyhteys on jumissa yli 10 sekuntia, komento päätetään, eikä haeta muutostietoja puppet-masterilta, muutoin haetaan muutokset puppet-masterilta. (Kuva 26.)




(Kuva 26. Puppet SSL virheenkierto skripti.)

Skriptikutsu lisättiin ajastettuihin ajoihin samalla tavalla kuin aikaisemmin oli suoraan puppet komento. (Kuva 27.)



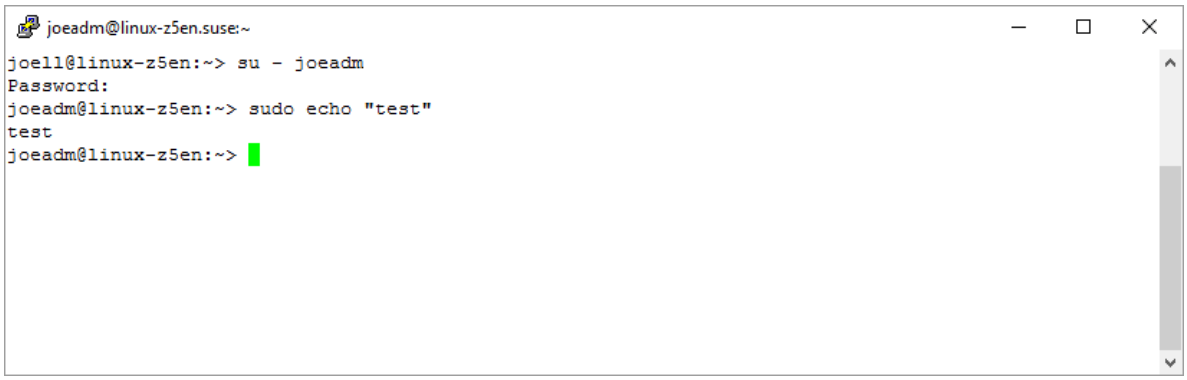

(Kuva 27. Puppet ajastus skripti lisättynä ajastettuihin ajoihin.)

Seuraavaksi vain odotettiin, mutta mitään ei tapahtunut. Tämä johtui siitä, että puppet agentissa on virhe, joka kaataa itsensä, jos puppet agent ajetaan ajastetusti. Tästä löytyy lisää tietoja puppetlabsin tikettijärjestelmästä. (Puppet Labs 2014.)

Lopulta tultiin siihen tulokseen, että puppet agent ajetaan aina käynnistyksen yhteydessä ja jätetään taustalle. Tämä hakee 30 minuutin välein muutoksia puppet-masterilta.

Seuraavaksi tulikin ongelmaksi se, ettei LDAP tunnukset toimineet, vaikka kaikki ohjelmat asentuivat ja konfiguraatiotiedostot korvautuivat virheittä. Järjestelmän lokeille tuli maininta siitä, ettei NSCD palvelu saisi olla samaan aikaan varastoimassa (caching) käyttäjätietoja SSSDn kanssa. Tämän takia lisättiin muutos ’ldap_services.pp’ luokkaan, jossa otetaan NSCD palvelu pois käytöstä päätekoneella. Jos NSCD –palvelu jostain syystä tulisi olla päällä päätekoneella, niin palvelun sulkemisen sijaan, voitaisiin suoraan korvata myös NSCD konfiguraatiotiedostot oikeanlaisiksi tarvittaessa.

Lopulta kaikki toimi oikein. Ensin kirjauduttiin admin-tason LDAP tunnuksella, ja ajettiin päätekoneella sudo -etuliitteellä testikomento. (Kuva 28.)




(Kuva 28. LDAP käyttäjän kirjautumisen ja pääkäyttäjäkomennon testaus.)

# 6 Pohdinta

Projektissa toteutettiin LDAP käyttöönotto käyttäen keskitetyn ohjelmistohallinnan työkalua Puppet, joka on mahdollista myös laajentaa tarvittaessa useampiin työpisteisiin. Tarkoituksena projektilla oli antaa yleisymmärrys projektintekijän omasta lähestymistavasta siihen, kuinka keskitettyä ratkaisua voidaan alkaa rakentamaan, sekä antaa lukijalle ymmärrystä yleisesti Puppetin toiminnasta.

Projektintekijä on hyvin tietoinen siitä, että on useampia lähestymistapoja Puppet arkkitehtuurin tekoon, projektin lähetymistapana oli suora tiedostojen korvaus, niiden konfiguroinnin sijaan. Ympäristöissä, joissa käytetään useampia eri LDAP yhteyksiä, tämä tarkoittaisi jokaisen erillisen yhteyden konfiguraatiotiedostojen ylläpitoa keskitetyllä ohjelmistonhallinnan palvelimella, ja niille omien manifestien ja moduulien tekoa, tällä arkkitehtuurimallilla.

Puppetin dokumentoinnit ovat erittäin kattavat, mutta ne saattavat aiheuttaa hyvinkin nopeasti hämmennystä yleisen arkkitehtuurin hahmottamisesssa, ja sen kanssa miten toimivaa järjestelmää halutaan alkaa rakentamaan. Tämä ajoikin projektintekijän tekemään tämän projektin, jotta saadaan suoraviivainen ja pelkistetty esimerkki hiukka monimutkaisemman puppet arkkitehtuurin rakentamisesta.

Vaikka projekti oli rajattu vain käsittelemään aihe SuSE Linux käyttöjärjestelmän perspektiivistä, niin puppetin yleisrakenne on aina sama, tuetusta käyttöjärjestelmästä riippumatta. Huomioitava on aina tietysti käyttöjärjestelmän omat rajoitteet tai poikkeavuudet toiminnassa. Vaikka yleisrakenne on sama, niin poikkeavaisuuksia löytyy aina. Kunnollisella testauksella yleisimmät ja mahdolliset virhetapaukset saadaan kiinni etukäteen.

Yhteenvetona projektista, itse päätavoite saatiin toteutettua, joka oli saada keskitetty käyttäjänhallinta keskitetysti levitettyä päätekoneelle, käyttämällä keskitettyä ohjelmistonhallintaa. Tähän vielä kehitysideana jää Puppetin yksinkertaistettu tai automatisoitu levitys päätekoneelle, tämänkin voisi toteuttaa SuSEn omalla autoyast esiasennus-skriptillä käyttöjärjestelmän asennuksen yhteydessä, mutta sen toteutus olisi ihan oma projekti.

## Lähteet


Fedorahosted 2016a, SSSD: Wiki. Luettavissa: https://fedorahosted.org/sssd/wiki. Luettu: 18.3.2016

Fedorahosted 2016b, SSSD: FAQ. Luettavissa: https://fedorahosted.org/sssd/wiki/FAQ. Luettu: 18.3.2016

Fedoraproject 2010, SSSD features. Luettavissa: https://docs.fedoraproject.org/en-US/Fedora/14/html/Deployment_Guide/sect-SSSD_User_Guide-Introduction-SSSD_Features.html. Luettu: 18.3.2016

LWN.net 2011, SSSD: System Security Services Daemon. Luettavissa: http://lwn.net/Articles/457415/. Luettu: 18.3.2016

OpenLDAP 2014, Release Road Map. Luettavissa: http://www.openldap.org/software/roadmap.html. Luettu 1.4.2016

OpenLDAP 2011a, Introduction to OpenLDAP Directory Services. Luettavissa: http://www.openldap.org/doc/admin24/intro.html. Luettu 1.4.2016

OpenLDAP 2011b, Security Considerations. Luettavissa: http://www.openldap.org/doc/admin24/security.html. Luettu 1.4.2016

Puppet Labs 2012a, Wiki. Luettavissa: https://projects.puppetlabs.com/projects/puppet/wiki. Luettu: 10.3.2016

Puppet Labs 2012b, Introduction. Luettavissa: http://docs.puppetlabs.com/guides/introduction.html. Luettu: 10.3.2016

Puppet Labs 2012c, Platforms. Luettavissa: https://docs.puppetlabs.com/guides/platforms.html. Luettu: 10.3.2016

Puppet Labs 2012d, Big Picture. Luettavissa: https://projects.puppetlabs.com/projects/puppet/wiki/Big_Picture. Luettu: 10.3.2016

Puppet Labs 2012e, Architechture. Luettavissa: http://docs.puppetlabs.com/puppet/3.7/reference/architecture.html. Luettu: 10.3.2016

Puppet Labs 2012f, Language: Basics. Luettavissa: http://docs.puppetlabs.com/puppet/latest/reference/lang_summary.html. Luettu 10.3.2016

Puppet Labs 2012g, Language: Classes. Luettavissa: https://docs.puppetlabs.com/puppet/latest/reference/lang_classes.html. Luettu 1.4.2016

Puppet Labs 2012h, Language: Resources. Luettavissa: https://docs.puppetlabs.com/puppet/latest/reference/lang_resources.html. Luettu 1.4.2016

Puppet labs 2012i, Background Reference: SSL and Related Topics. Luettavissa: https://docs.puppetlabs.com/background/ssl/index.html. Luettu 1.4.2016

Puppet Labs 2014, PUP-1320: ”Caught TERM; calling stop”. Luettavissa: https://tickets.puppetlabs.com/browse/PUP-1320. Luettu 10.2.2016

Turnbull, J. 2008, Pulling Strings with Puppet: Configuration Management Made Easy. Apress. New York.

Ubuntu Wiki 2014, SSSD. Luettavissa: https://wiki.ubuntu.com/Enterprise/Authentication/sssd. Luettu: 18.3.2016.

Wikipedia 2016, List of LDAP software. Luettavissa: https://en.wikipedia.org/wiki/List_of_LDAP_software. Luettu 1.4.2016.

## Liitteet

### Liite 1. /etc/puppet/manifests/site.pp

```
#/etc/puppet/manifests/site.pp
node default {
}

node 'linux-z5en.suse' {
include sled12ldap
}
```

### Liite 2. /etc/puppet/modules/sled12ldap/manifests/init.pp

```
# /etc/puppet/modules/sled12ldap/manifests/init.pp
class sled12ldap {
  include ldap_packages
  include ldap_configuration
  include ldap_services
}
```

### Liite 3. /etc/puppet/modules/sled12ldap/manifests/ldap_packages.pp

```
# /etc/puppet/modules/sled12ldap/manifest/ldap_packages.pp
class ldap_packages {
$packages = [ 'openldap2-client', 'yast2-ldap', 'nss_ldap', 'pam_ldap', 'sssd-ldap', 'sssd' ]
package { $packages: ensure => 'installed' }
}
```

**Liite 4. /etc/puppet/modules/sled12ldap/manifests/ldap_configuration.pp**

```
# /etc/puppet/modules/sled12ldap/manifests/ldap_configuration.pp
class ldap_configuration {
$modulehome = "puppet:///modules/sled12ldap"


file {  "/etc/nsswitch.conf" :
                source => "$modulehome/nsswitch.conf",
                mode => '0644',
                owner => 'root',
                group => 'root',
                }

file {  "/etc/pam.d/common-session" :
                source => "$modulehome/pam/common-session",
                mode => '0644',
                owner => 'root',
                group => 'root',
                require => Package['sssd'],
                }

file {  "/etc/pam.d/common-password" :
                source => "$modulehome/pam/common-password",
                mode => '0644',
                owner => 'root',
                group => 'root',
                require => Package['sssd'],
                }

file {  "/etc/pam.d/common-auth" :
                source => "$modulehome/pam/common-auth",
                mode => '0644',
                owner => 'root',
                group => 'root',
                require => Package['sssd'],
                }
```

```
file {  "/etc/pam.d/common-account" :
                source => "$modulehome/pam/common-account",
                mode => '0644',
                owner => 'root',
                group => 'root',
                require => Package['sssd'],
                }

file {  "/etc/sssd/sssd.conf" :
                source => "$modulehome/sssd/sssd.conf" ,
                mode => '0600',
                owner => 'root',
                group => 'root',
                require => Package['sssd'],
                }

file { "/etc/sudoers" :
                source => "$modulehome/sudoers",
                mode => '0440',
                owner => 'root',
                group => 'root',
                }

}
```

**Liite 5. /etc/puppet/modules/sled12ldap/manifests/ldap_services.pp**

```
# /etc/puppet/modules/sled12ldap/manifests/ldap_services.pp

class ldap_services {
                service { "sssd":
                ensure => running,
                enable => true,
                subscribe => File["/etc/sssd/sssd.conf"],
                }
                service {”nscd”:
                ensure => stopped,
                enable => false,
                }
}
```

**Liite 6. /etc/puppet/modules/sled12ldap/files/nsswitch.conf**

```
#
# /etc/nsswitch.conf
#
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# The entry '[NOTFOUND=return]' means that the search for an
# entry should stop if the search in the previous entry turned
# up nothing. Note that if the search failed due to some other reason
# (like no NIS server responding) then the search continues with the
# next entry.
#
# Legal entries are:
#
#       compat              Use compatibility setup
#       nisplus             Use NIS+ (NIS version 3)
#       nis                 Use NIS (NIS version 2), also called YP
#       dns                 Use DNS (Domain Name Service)
#       files               Use the local files
#       [NOTFOUND=return]       Stop searching if not found so far
#
```

*# For more information, please read the nsswitch.conf.5 manual page.*
*#*

*# passwd: files nis*
*# shadow: files nis*
*# group:  files nis*

*passwd: compat sss*
*group:  compat sss*

*hosts:  files mdns_minimal [NOTFOUND=return] dns*
*networks:  files dns*

*services:  files*
*protocols:  files*
*rpc:  files*
*ethers: files*
*netmasks:  files*
*netgroup:  files nis*
*publickey:  files*

*bootparams:  files*
*automount:  files nis sss*
*aliases:  files*
*passwd_compat:  files*
*group_compat:  files*
*sudoers:  files sss*

**Liite 7. /etc/puppet/modules/sled12ldap/files/pam/common-session**

```
#%PAM-1.0
#
# This file is autogenerated by pam-config. All changes
# will be overwritten.
#
# Session-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define tasks to be performed
# at the start and end of sessions of *any* kind (both interactive and
# non-interactive
#
session  optional       pam_mkhomedir.so
session required        pam_limits.so
session required        pam_unix.so     try_first_pass
session optional        pam_sss.so
session optional        pam_umask.so
session optional        pam_systemd.so
session optional        pam_gnome_keyring.so    auto_start only_if=gdm,gdm-
password,lxdm,lightdm
session optional        pam_env.so
```

**Liite 8. /etc/puppet/modules/sled12ldap/files/pam/common-password**

```
#%PAM-1.0
#
# This file is autogenerated by pam-config. All changes
# will be overwritten.
#
# Password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define  the services to be
# used to change user passwords.
#
password        requisite       pam_cracklib.so
password        optional        pam_gnome_keyring.so    use_authtok
password        sufficient      pam_unix.so     use_authtok nullok shadow try_first_pass
password        required        pam_sss.so      use_authtok
```

**Liite 9. /etc/puppet/modules/sled12ldap/files/pam/common-auth**

```
#%PAM-1.0
#
# This file is autogenerated by pam-config. All changes
# will be overwritten.
#
# Authentication-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use the
# traditional Unix authentication mechanisms.
#
auth    required        pam_env.so
auth    optional        pam_gnome_keyring.so
auth    sufficient      pam_unix.so     try_first_pass
auth    required        pam_sss.so      use_first_pass
```

**Liite 10. /etc/puppet/modules/sled12ldap/files/pam/common-account**

```
#%PAM-1.0
#
# This file is autogenerated by pam-config. All changes
# will be overwritten.
#
# Account-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the accountorization modules that define
# the central access policy for use on the system.  The default is to
# only deny service to users whose accounts are expired.
#
account requisite       pam_unix.so     try_first_pass
account sufficient      pam_localuser.so
account required        pam_sss.so      use_first_passLiite 11.
```

**Liite 11. /etc/puppet/modules/sled12ldap/files/sssd/sssd.conf**

```
[sssd]
config_file_version = 2
services = nss, pam, sudo
domains = puppet.com
[nss]
filter_users = root
filter_groups = root
pam]
[domain/puppet.com]
id_provider = ldap
auth_provider = ldap
ldap_schema = rfc2307bis
cache_credentials = true
ldap_uri = ldap://puppet.com
ldap_search_base = dc=puppet,dc=com
ldap_tls_reqcert = never
sudo_provider = ldap
ldap_sudo_search_base = ou=sudoers,dc=puppet,dc=com
```

**Liite 12. /etc/puppet/modules/sled12ldap/files/sudoers**

```
## sudoers file.
##
## This file MUST be edited with the 'visudo' command as root.
## Failure to use 'visudo' may result in syntax or file permission errors
## that prevent sudo from running.
##
## See the sudoers man page for the details on how to write a sudoers file.
##

##
## Host alias specification
##
## Groups of machines. These may include host names (optionally with wildcards),
## IP addresses, network numbers or netgroups.
# Host_Alias    WEBSERVERS = www1, www2, www3

##
## User alias specification
##
## Groups of users.  These may consist of user names, uids, Unix groups,
## or netgroups.
# User_Alias    ADMINS = millert, dowdy, mikef

##
## Cmnd alias specification
##
## Groups of commands.  Often used to group related commands together.
# Cmnd_Alias    PROCESSES = /usr/bin/nice, /bin/kill, /usr/bin/renice, \
#                           /usr/bin/pkill, /usr/bin/top

##
## Defaults specification
##
## Prevent environment variables from influencing programs in an
## unexpected or harmful way (CVE-2005-2959, CVE-2005-4158, CVE-2006-0151)
Defaults always_set_home
```

*## Path that will be used for every command run from sudo*
*Defaults secure_path="/usr/sbin:/usr/bin:/sbin:/bin"*
*Defaults env_reset*
*## Change env_reset to !env_reset in previous line to keep all environment variables*
*## Following list will no longer be necessary after this change*

*Defaults env_keep = "LANG LC_ADDRESS LC_CTYPE LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE LC_TIME LC_ALL LANGUAGE LINGUAS XDG_SESSION_COOKIE"*
*## Comment out the preceding line and uncomment the following one if you need*
*## to use special input methods. This may allow users to compromise the root*
*## account if they are allowed to run commands without authentication.*
*#Defaults env_keep = "LANG LC_ADDRESS LC_CTYPE LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE LC_TIME LC_ALL LANGUAGE LINGUAS XDG_SESSION_COOKIE XMODIFIERS GTK_IM_MODULE QT_IM_MODULE QT_IM_SWITCHER"*

*## Do not insult users when they enter an incorrect password.*
*Defaults !insults*

*##*
*## Uncomment to enable logging of a command's output, except for*
*## sudoreplay and reboot. Use sudoreplay to play back logged sessions.*
*# Defaults log_output*
*# Defaults!/usr/bin/sudoreplay !log_output*
*# Defaults!/sbin/reboot !log_output*

*## In the default (unconfigured) configuration, sudo asks for the root password.*
*## This allows use of an ordinary user account for administration of a freshly*
*## installed system. When configuring sudo, delete the two*
*## following lines:*
*#Defaults targetpw # ask for the password of the target user i.e. root*
*#ALL ALL=(ALL) ALL # WARNING! Only use this together with 'Defaults targetpw'!*

*##*
*## Runas alias specification*

```
##

##
## User privilege specification
##
root ALL=(ALL) ALL
## Uncomment to allow members of group wheel to execute any command
# %wheel ALL=(ALL) ALL

## Same thing without a password
# %wheel ALL=(ALL) NOPASSWD: ALL

## Read drop-in files from /etc/sudoers.d
## (the '#' here does not indicate a comment)
#includedir /etc/sudoers.d
```